输入排队中抢占式的短包优先调度算法

李文杰, 刘 斌
(清华大学计算机科学与技术系, 北京 100084)

摘 要: 调度算法决定了输入排队交换结构的性能. 本文根据 Internet 业务特征提出调度算法应保证短包的高优先级和低延迟. 已有包方式调度中, 长包信元的连续传输将造成短包长时间等待. 为解决该问题, 本文设计了一种低复杂度抢占式交换结构, 并提出了相应的抢占式短包优先调度算法(P-SPF). 短包优先可减小 TCP 流的 RTT, 并由此提高 TCP 之性能. 通过排队论分析和实际业务源模型下仿真可知: P-SPF 取得短包近似为零的平均包等待时间, 同时达到 94% 的系统吞吐量.

关键词: 输入排队; 包方式; 抢占式; 短包优先
中图分类号: TP393.05 文献标识码: A 文章编号: 0372-2112 (2005) 04-0577-07

Preemptive Short-Packet-First Scheduling in Input Queueing Switches

LI Wen-jie, LIU Bin
(*Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China*)

Abstract: Scheduling algorithms make a great impact on the performance of input queueing switches. From Internet traffic characteristics, it is pointed out that short packets should be guaranteed higher priority and lower delay in scheduling algorithms. In general packet-mode scheduling, short packets suffer from long waiting time due to the continuous transferring of cells of long packets. To solve this problem, we study the low-complexity preemptive packet-mode scheduling and propose the algorithm called preemptive short packets first (P-SPF). P-SPF improves the TCP performance by means of the fact that round-trip delays of TCP flows are greatly reduced. Both the analysis in queueing theory and the simulation results with respect to a real traffic model show that P-SPF can achieve almost zero average packet waiting time for short packets, while keeping a high overall throughput up to 94%.

Key words: input queueing switches; packet-mode; preemptive; short packets first

1 引言

输入排队交换结构被广泛应用于高速核心路由器中, 如 Cisco 12000 系列[1], Tiny-Tera[2] 和 BBN[3]. 同输出排队和共享缓存相比, 输入排队存储带宽低, 硬件实现简单. 同时, 虚拟输出排队(VOQ) 解决了其队头阻塞问题(HOL)[4], 使得输入排队调度算法达到 100% 系统吞吐量. 输入排队调度算法一般都以时隙为单位来调度固定长度的信元, 时隙就是交换网络传送一个信元所对应的时间. 根据对信元调度方式的不同, 可以将输入排队调度算法分为两大类: 信元方式调度和包方式调度.

(1) 信元方式调度: IP 包到达输入端口后, 将被分割成相互独立的信元. 每个时隙中, 调度算法以信元为粒度进行调度并重新配置交换网络. 在输出端口, 信元被重组回一个完整的 IP 包. 由于从不同输入端口到达的属于不同包的信元会被间隔传送, 因此需要使用虚拟输入排队(VIQ) 来实现信元重组. 所谓 VIQ, 就是在每个输出端口中, 为每个输入端口分别分配虚拟队列并顺序缓存来自对应输入端口的信元. 信元方式调度已被广泛应用, 典型的调度算法如 *i*SLIP[5], *i*LPF[6] 和 DRRM[7], 但其存在如下不足: (a) 由于信元之间的相互独立性, 每个信元都需要目的输出端口标签, 这会造成交换网络效率下降; (b) 当路由器端口数很多, 或支持超长 IP 包时, 信元重组将占用大量存储器.

(2) 包方式调度: Marsan M A 等人提出了包方式调度[8]. 在包方式调度中, 同一个包的所有信元被当作一个原子调度实体, 调度算法保证匹配的输入输出端口连续得到服务直到属于同一包的所有信元都被传送完毕. 关于包方式调度的主要最新研究成果有: 文[8] 证明了当输入为 Bernoulli 独立同分布到达过程时, 最大权重匹配调度算法是稳定的; 文[9] 将该结论扩展到了输入是一般可控的更新过程; 文[10] 进一步表明包方式调度算法也可保证各个流的公平性. 值得注意的是, 属于同一包的所有信元是连续传输的, 因此只有头信元需要目的输出端口标签, 在输出端口已不再需要信元重组, 这就去掉了由信元重组而产生的附加延迟. 总之, 同信元方式调度相

收稿日期: 2004-08-11; 修回日期: 2004-11-24
基金项目: 国家"863"高科技研究发展计划(No. 2002AA103011-1; 2003AA115110); 国家自然科学基金(No. 60173009; 60373007)

比,包方式调度更适合于路由器中变长 IP 包交换.

然而,从传输层协议的角度来看,包方式调度会引起TCP性能的下降.为说明该问题,我们从协议和包长分布两方面研究了 Internet 业务特征.分析对象是一个从 Internet 上截获的流量 Auckland-II[11],结果显示:从包的个数来看,TCP 包占 86.5%,UDP 包占12.8%,其它协议的包只占 0.7%,由此可见 TCP 是 Internet 中最主要的传输层协议.图 1 给出了 Internet 累计包长分布,由此图可知在 Internet 中长度不超过 64 字节的包约占所有包的 60%.这些包大部分是 TCP 的 ACK 包,或其它 TCP 的控制包,如 SYN,FIN 或 RST 等.这些短包的延迟直接影响着 TCP 的 RTT 时间,而 RTT 对 TCP 性能有着非常大的影响[12].路由器中调度算法的最终目的是为上层传输协议提供更好的服务,因此一个有效的调度算法应尽量减小这些短包的延迟来取得更好的TCP性能.而本文的仿真结果表明:已有包方式调度中长包信元的连续传输将造成短包阻塞并使其延迟增大.

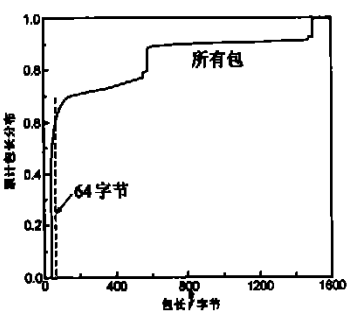

图 1 Internet累计包长分布

解决上述问题就需要提高短包服务的优先级,然而传统区分服务模型[13]并不适用于此.区分服务模型针对的是不同优先级流之间的调度,它需要各个终端设置每个数据包的优先级.而本文所提出并解决的是 TCP 流内不同长度包之间的调度,是从交换的角度来研究如何提高 TCP 性能.同时,本文所提出的方法不需要终端做任何设置或修改.包方式调度是在信元方式调度的基础上引入了"包"的概念,因此可看作信元方式调度的扩展.在此,我们在包方式调度中引入抢占操作来解决已有包方式调度中短包阻塞问题.同文[14]和文[15]中区分服务不同长度流的思想类似,本文应用了排队论理论:优先服务那些短服务时间的包可降低所有包的平均包等待时间[16].为此,我们研究了抢占模式下的包方式调度结构,并提出了抢占式的短包优先调度算法(P-SPF).根据 Internet 中短包的特点:数量多但到达率低,P-SPF 将短包缓存到一个独立 FIFO 中,然后通过中断长包传输来优先传送短包.通过理论分析和性能仿真得到:P-SPF 可取得短包近似为零的平均包等待时间,并由此降低所有包的平均包等待时间,同时,P-SPF 的系统吞吐量可达 94%.

本文结构如下:第 2 部分介绍了抢占式的交换结构和 P-SPF 调度算法;第 3 部分分析了平均包等待时间和 P-SPF 的最大系统吞吐量;第 4 部分给出了实际业务源模型下的仿真,并分析了相应实验结果;最后,第 5 部分对全文作了总结.

2 抢占式的交换结构和 P-SPF 调度算法

作为交换网络效率和调度算法运行时间的折衷,本文取信元长度为 64 字节(该长度只为信元净荷长度,不包括其它开销).信元越小,则 IP 包分割中由最后一个信元填充所造成的带宽浪费就越少;反之,信元越大,时隙就越长,越有利于在硬件上实现调度算法.如果优先服务所有不大于 64 字节的包,则由短包和长包组成的混合TCP流内可能会发生包的乱序.乱序在路由器中是允许的[17],而且由于传输链路并行和路由器内部模块的并行处理等使得乱序现象在 Internet 中的确存在[18].尽管如此,本文仍提出了一种更详细区分长包和短包的标准来保证同一 TCP 流内数据包的顺序.

区分长包和短包的标准:一个 IP 包被划分为短包当且仅当:(1)它的包长不大于 64 字节;(2)它不是 TCP 包,或者是不包含任何数据的TCP 包.除此之外,所有其它的包都被划分为长包.

该区分长包和短包的标准可以很容易地在网络处理器中实现,图 2 给出了一个实现流程.根据数据链路层的协议字段可判断出封装的是否是 IP 包,如是,则 IP 包长、IP 头长和 IP 协议域均可从 IP 报头中提取出来.如果 IP 报头中的协议域指示的是 TCP 包,则进一步可从 TCP 的头部字节中提取 TCP 头长.如果 IP 总长等于 IP 头长加上 TCP 头长,则此 TCP 包不含任何数据.

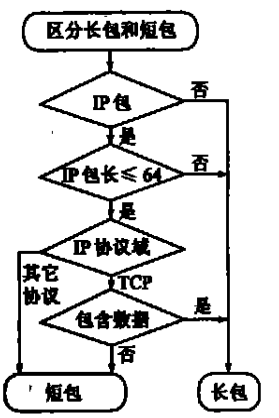

图 2 区分长包和短包标准的实现流程

从 Internet 业务流量 Auckland-II 的分析中可知:在所有不超过 64 字节的包中,包含数据的 TCP 包只占 3.9%,其它包则占 96.1%.也就是说,使用区分标准所得短包占所有不超过 64 字节包的 96.1%.由此,在后面的分析和仿真中我们作了一种比较精确的近似:将所有不大于 64 字节的包都作为短包来处理,即长度为一个信元的包对应短包,大于一个信元的对应长包,这样就不需要考虑传输层协议并建立一个第四层的业务源模型来进行分析和仿真.

2.1 抢占式的交换结构

图 3 给出了抢占式的交换结构,其中参数 N 是该交换结构的端口数,在核心路由器中一般为 16,32 或者更多,S 和 L 分别表示短包和长包.该结构和文[8]中包方式调度结构非常相似,不同的是在输入/输出端口分别分配了用于服务短包的独立 FIFO 队列.

一个包到达输入端口后,将被分割为固定长度的信元.输入缓存单元根据包的类型(长包或短包)把它缓存到相应的队列中:如果是一个短包,则直接缓存到 IFIFO_S 中;如果是一个长包,则缓存到相应的 VOQ_K 中,其中 K 为该包的目的输出端口号.输入调度单元监控短包队列和长包队列的状态,并向交换调度单元同时发送短包和长包的输出请求.当收到交换调度单元的仲裁确认后,传送被确认队列中的队头信元.

交换调度单元在每个时隙中执行 P-SPF 调度算法,重新配置交换网络,并将仲裁结果发送到各个输入端口,允许被确认的输入端口在下一个时隙中传送信元.

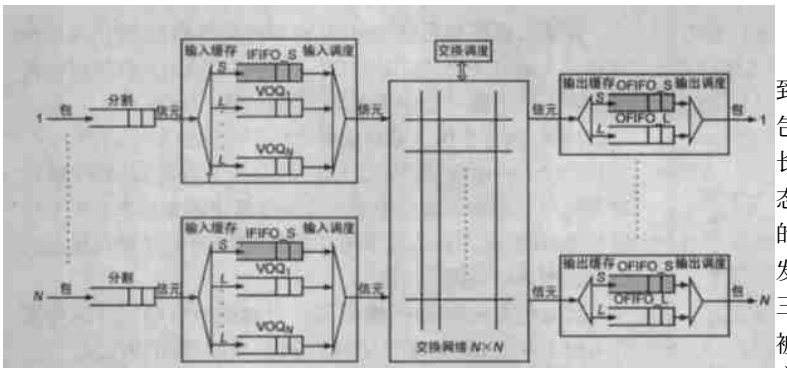

图 3　抢占式的交换结构

信元到达输出端口后, 输出缓存单元将短包和长包的信元分别缓存到 OFIFO_S 和 OFIFO_L 中. 当输出端口缓存了完整短包或完整长包后, 输出调度单元则负责传送 OFIFO_S 中的短包和 OFIFO_L 中的长包到输出链路. 在输出链路空闲时, 输出调度单元优先选择传送 OFIFO_S 中的短包. 而在向输出链路传送包的过程中, 输出调度单元不可中断包的传输.

2.2　P-SPF 的交互式调度过程

P-SPF 属于交互式调度算法, 在每个时隙中, 包括并行的两个调度过程: 短包调度和长包调度. 每个输入端口 $i(1 \leqslant i \leqslant N)$ 有一个长包调度的指针 $IP_L(i)$, 每个输出端口 $j(1 \leqslant j \leqslant N)$ 有一个长包调度的指针 $OP_L(j)$ 和一个短包调度的指针 $OP_S(j)$.

(1) 短包调度过程

每个短包只有一个信元, 每个输入端口只分配一个队列缓存短包, 而且短包调度可抢占长包传输, 即不需要考虑输入和输出端口中的长包传送状态, 故短包调度非常简单, 共包括如下两步:

步骤一: 输入请求

每个输入端口向其 IFIFO_S 中队头信元的目的输出端口发送一个短包请求.

步骤二: 输出确认

每个输出端口 j 收到短包请求后, 就从短包指针 $OP_S(j)$ 所指的输入端口开始轮询, 在遇到第一个发送了短包请求的输入端口后就确认该输入端口, 同时将指针更新到该被确认输入端口的下一个端口. 由于每个输入端口只向一个输出端口发送短包请求, 因此输出端口的确认总会被输入端口接受.

(2) 长包调度过程

在长包调度的过程中, 每个输入/输出端口在每个时隙有两种状态:

- 空状态: 没有长包信元在这个状态传送, 或者长包的最后一个信元正在传送;
- 忙状态: 输入/输出端口正在传送一个长包的头信元或中间信元.

长包调度包括如下三步交互过程:

步骤一: 输入请求

每个空状态的输入端口向所有其缓存长包的目的输出端口发送长包请求, 每个忙状态的输入端口不发送任何长包请求.

步骤二: 输出确认

每个输出端口 j 可概括为如下四种情况: (1) 收到短包请求, 则不确认任何长包请求; (2) 未收到短包请求, 处于忙状态, 则仍然确认前面时隙中建立的长包匹配输入端口; (3) 未收到短包请求, 处于空状态, 且收到长包请求, 则从长包指针 $OP_L(j)$ 所指向的输入端口开始, 采用轮询的方式确认首先遇到的发送了长包请求的输入端口, 如果这个确认在步骤三中被输入端口接受, 则将长包指针 $OP_L(j)$ 更新到被确认输入端口的下一个端口, 否则保持该指针不变; (4) 于空状态, 且未收到任何短包和长包请求, 则不确认任何输入端口.

步骤三: 输入接受

每个输入端口 i 在收到多个输出端口的确认后, 从长包指针 $IP_L(i)$ 所指的输出端口开始采用轮询的方式接受首先遇到输出端口的长包确认, 并同时更新长包指针 $IP_L(i)$ 到被接受输出端口的下一个端口.

在输出端口确认了某输入端口的长包请求, 并被该输入端口接受后, 新匹配的输入/输出端口均由空状态变为忙状态. 当输入端口在发送一个长包的最后信元, 输出端口在接收一个长包的最后信元时, 相应的输入/输出端口更新到空状态.

2.3　P-SPF 的性质

P-SPF 具有如下性质:

性质 1: 短包被赋予高优先级并可抢占长包传输, 通过后面的分析和仿真可知: 通过抢占可取得近乎为零的短包平均包等待时间, 并可降低所有包的平均包等待时间.

性质 2: 对短包的服务不存在饿死现象. 这是由于短包被缓存到单独的 IFIFO_S 队列中, 并采用轮询方式服务, 因此处于 IFIFO_S 队头位置的短包最多等 N 个时隙就会得到服务.

性质 3: 同已有包方式调度相比, P-SPF 只略微增加了空间复杂度, 而不增加时间复杂度. 这是因为短包调度和长包调度是在调度单元中并行执行的, 而长包调度比短包调度复杂, 因此是长包调度决定了 P-SPF 的调度完成时间. 同时, P-SPF 中的长包调度和已有包方式调度的时间复杂度是相同的, 因此 P-SPF 并不增加已有包方式调度的时间复杂度.

性质 4: 在完成短包调度和长包调度后, 一个输入端口可能会同时接受一个短包确认和一个长包确认, 这种情况下, 输入端口优先传送短包, 然后在后面时隙中没有短包传送时马上开始传送长包. 同样, 一个输出端口在收到一个短包请求时, 会优先确认短包请求, 则原来已经匹配的输入端口的长包请求会被暂时挂起.

图 4 给出了一个四端口交换网络中 P-SPF 的调度例子. 所有输入和输出端口初始都处于空状态, 所有指针都指向端口 1. 在步骤一中, 输出端口 1 仅收到短包请求, 输出端口 2 和 3 仅收到长包请求, 输出端口 4 同时收到长包和短包请求. 在步骤二中, 输出端口 1 确认输入端口 1 的短包请求, 输出端口 2 和 3 分别确认输入端口 2 和 3 的长包请求, 输出端口 4 确认输入端口 3 的短包请求. 在步骤三中, 输入端口 1 接受短包确

认, 输入端口 2 接受长包确认, 输入端口 3 同时接受来自输出端口 3 的长包确认和来自输出端口 4 的短包确认, 输入端口 4 则没有收到任何确认. 最后, 更新所有匹配输入端口(1, 2, 3)和输出端口(1, 2, 3, 4)中的相应长包调度和短包调度指针.

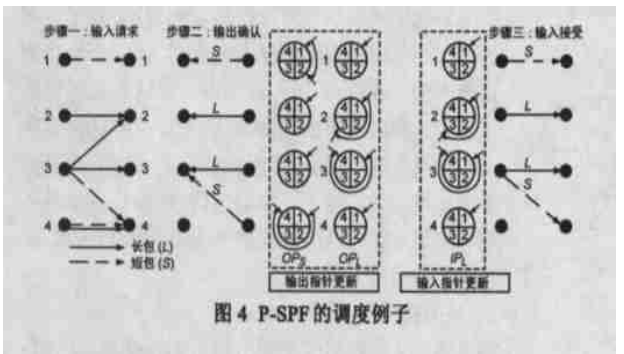

图 4 P-SPF 的调度例子

3 性能分析

3.1 平均包等待时间分析

首先, 我们定义如下关于延迟的参数:

- 包到达时间: 包的最后一个信元到达输入端口的时间;
- 包离开时间: 包的最后一个信元到达输出端口的时间;
- 包延迟: 包离开时间减去包到达时间;
- 包服务时间: 交换网络用于传送一个包的时间;
- 包等待时间: 包延迟减去包服务时间, 也就是包在输入队列中的等待时间.

无论是已有包方式调度中, 还是抢占模式下, 同一个包的服务时间都是相同的, 它只和包长相关, 因此本文主要考虑包等待时间, 该参数直接反应了包延迟.

然后, 我们任选一个输出端口, 研究它对所有去往该输出端口包的服务过程, 并定义如下排队参数:

(1) 包到达率: λ_S, λ_l 和 λ 分别表示短包、长包和所有包的到达率, 它们有如下关系:

$$\lambda = \lambda_S + \lambda_l \tag{1}$$

(2) 包服务率: μ_S, μ_l 和 μ 分别表示短包、长包和所有包的服务率, 则

$$\frac{1}{\mu} = \frac{\lambda_S}{\lambda} \times \frac{1}{\mu_S} + \frac{\lambda_l}{\lambda} \times \frac{1}{\mu_l} \tag{2}$$

(3) 负载: 包到达率与包服务率之比. ρ_s, ρ_l 和 ρ 分别表示短包、长包和所有包的负载, 由式(2)可得

$$\rho = \rho_S + \rho_l \tag{3}$$

(4) 平均包服务时间: $E(S_S)$, $E(S_l)$ 和 $E(S)$ 分别表示短包、长包和所有包的平均包服务时间.

(5) 包服务时间方差系数: 包服务时间均方差与其期望之比, 用 C_V 表示.

(6) 平均包等待时间: $E(W_S)$, $E(W_l)$ 和 $E(W_P)$ 分别表示抢占模式下短包、长包和所有包的平均包等待时间.

(7) 抢占增益: 已有包方式中的平均包等待时间 $E(W_G)$ 和抢占模式下的平均包等待时间 $E(W_P)$ 之比, 用 G 表示:

$$G = \frac{E(W_G)}{E(W_P)} \tag{4}$$

在输入排队交换结构中, 存在两种类型冲突: 输出端口冲突和输入端口冲突. 输出端口冲突: 当一个输出端口同时收到来自多个输入端口的请求时, 它只能确认一个输入端口; 输入端口冲突: 当一个输入端口同时收到多个输出端口的确认时, 它只能接受一个确认. 同文[8]中的分析过程类似, 在中低负载下我们忽略输入端口冲突并应用排队论的方法来分析平均包等待时间, 而在高负载下考虑输入端口冲突并建立精确的仿真模型来研究延迟性能.

已有包方式调度中, 属于同一个包的所有信元连续传输且不被中断, 因此由 M/G/1 FCFS 排队模型[16]可得

$$E(W_G) = \frac{(1 + C_V^2)\rho E(S)}{2(1-\rho)} \tag{5}$$

在抢占模式包方式调度中, 由于短包被缓存到一个独立 FIFO 队列中, 并能抢占长包的传输, 因此系统中的长包对短包的服务过程没有任何影响. 短包的服务模型是一个 FIFO 队列的输入排队模型, 这和文[4]中的模型完全一致. 输入排队模型很难给出一个明确的解析解, 但从文[4]中的仿真可看出, 在信元到达率比较低的情况下, 输入排队和输出排队的性能是非常相近的. 输出排队下的平均包等待时间 W 为:

$$W = \frac{N}{N-1} \times \frac{p}{2(1-p)} \tag{6}$$

其中 p 是一个信元到达某个指定输出端口的概率.

短包服务模型中, $\mu_S = 1$, 则 $\rho_S = \lambda_S$, 并且在 Internet 中, 短包的到达率 λ_S 很小(见第 4 部分仿真业务源模型中的式(15)), 因此可由式(6)得到

$$E(W_S) = \frac{N-1}{N} \times \frac{\lambda_S}{2(1-\rho_S)} \tag{7}$$

当 $N \to \infty$, $E(W_S) = \dfrac{\lambda_S}{2(1-\rho_S)}$ (8)

对长包, 根据抢占式优先服务排队模型的结果[16]可得到

$$E(W_l) = \frac{1}{1-\rho_S} \times \left[E(S_l) + \frac{\lambda_S E(S_S^2) + \lambda_l E(S_l^2)}{2(1-\rho)} \right] - E(S_l) \tag{9}$$

因此, 由式(8)和式(9)可得到抢占模式下的平均包等待时间

$$E(W_P) = \frac{\lambda_S}{\lambda} E(W_S) + \frac{\lambda_l}{\lambda} E(W_l)$$

$$= \frac{1}{\lambda(1-\rho_S)} \left[\frac{1}{2}\lambda_S^2 + \rho_S\rho_l + \lambda_l \frac{\lambda E(S^2)}{2(1-\rho)} \right] \tag{10}$$

由式(5)和式(10)可得抢占增益 G:

$$G = \frac{(1 + C_V^2)(1-\rho_S)\rho^2}{(1-\rho)\lambda_S^2 + 2(1-\rho)\rho_S\rho_l + \lambda_l\lambda E(S^2)} \tag{11}$$

3.2 P-SPF 的最大系统吞吐量

短包的抢占将会挂起在前面时隙中已经匹配的长包, 图 5 给出了一个三端口交换网络例子. 其中, 在前一时隙中建立了三个长包匹配, 而当前时隙中输入端口 1 到达了一个去往输出端口 1 的短包, 则输入端

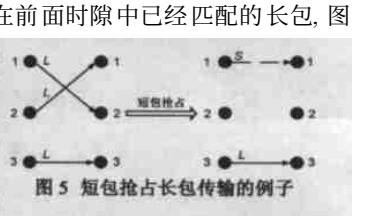

图 5 短包抢占长包传输的例子

口 1 和输出端口 1 之间将建立一个短包匹配, 同时输入端口 1 和输出端口 2、输入端口 2 和输出端口 1 之间的长包匹配将被挂起. 此时, 输出端口 1 有短包到达, 其带宽不会浪费, 而输出端口 2 没有短包到达, 长包挂起会使其被阻塞并造成带宽浪费.

我们任选一输出端口 O_B, 在所有输入端口均处于满负载时, 长包挂起造成的带宽浪费使得输入端口中总有去往 O_B 的长包. 在此, 我们记 O_B 当前长包匹配的输入端口为 I_A, 并定义如下两个概率事件:

(1) $A_s(O_B)$: 有一个去往输出端口 O_B 的短包到达某输入端口;

(2) $A_s(I_A)$: 有一个短包到达输入端口 I_A.

则输出端口 O_B 被阻塞的概率为:

$$\begin{aligned} P_r(O_B \text{ 被阻塞}) &= P_r(A_s(I_A) \cap \overline{A_s(O_B)}) \\ &= P_r(\overline{A_s(O_B)} \mid A_s(I_A)) \times P_r(A_s(I_A)) \\ &= \left(1-\frac{\lambda_S}{N}\right)^{N-1} \times \left(1-\frac{1}{N}\right) \times \lambda_S \end{aligned} \tag{12}$$

当 $N \to \infty$, 我们可得: $p_r(O_B \text{ 被阻塞}) = \lambda_S e^{-\lambda_S}$ (13)

设 T_{max} 为输出端口 O_B 能够取得的最大吞吐量, 则由式 (12) 和式 (13) 可得

$$\begin{aligned} T_{max} &= 1 - P_r(O_B \text{ 被阻塞}) \\ &= \begin{cases} 1-\lambda_S\left(1-\frac{1}{N}\right)\left(1-\frac{\lambda_S}{N}\right)^{N-1} & N < \infty \\ 1-\lambda_S e^{-\lambda_S} & N = \infty \end{cases} \end{aligned} \tag{14}$$

4 仿真实验

仿真的交换网络规模为 16×16 (N = 16), 信元长度为 64 字节. 共仿真了 1,000,000 个时隙, 其中处于稳态的从时隙 200,000 开始到时隙 800,000 结束的结果用于分析. 仿真业务源是两状态的 ON-OFF 模型.

OFF 状态: 在该状态没有包到达. OFF 状态服从几何分布, 并以一固定概率 q ($0 < q \leqslant 1$) 结束进入 ON 状态. OFF 状态的平均时隙数为 $1/q - 1$, 其中 q 等于 1 对应于输入端口处于满负载的情况.

ON 状态: 在这个状态有一个包到达. 当包的所有信元都到达后, ON 状态结束并进入 OFF 状态. 包长服从三元分布 TRIMODEL(a, b, c, P_a, P_b), 即以 P_a 概率等于 a 个信元, 以 P_b 概率等于 b 个信元, 或以 $1-P_a-P_b$ 概率等于 c 个信元. 同时, 到达包以 $1/N$ 的概率去往任一输出端口.

在仿真中, 包长分布 TRIMODEL 模型取如下参数: a = 1, b = 9, c = 24, P_a = 0.559 以及 P_b = 0.200, 即对应包长分别为 64, 576 和 1536 字节. 上述参数符合图 1 中关于包长分布的结果, 并和关于 Internet 业务量特征的文献[19, 20] 中的研究结果相一致. 因此模型 TRIMODEL(1, 9, 24, 0.559, 0.200) 比较真实地反应了 Internet 的包长分布.

在已有包方式调度中, 仿真的算法为硬件可实现的 4-iSLIP[5], 以及经典的理论算法: 最大尺寸匹配 (MSM)、以队列长度为权值的最大权重匹配 (MWM-QL) 和以队头信元年龄为权值的最大权重匹配 (MWM-CA)[21]. 同时, 我们修改了这些理论算法使其工作在抢占模式下, 即使用相应算法先调度短包, 然后在剩下的没有匹配的输入和输出端口中, 再使用相应算法调度长包, 分别记对应的抢占模式下算法为: P-MSM, P-MWM-QL 和 P-MWM-CA. P-SPF 对应于 4-iSLIP 在抢占模式下的硬件可实现算法.

由 TRIMODEL 模型可计算出输入端口在满负载时的短包到达率

$$\lambda_S = \frac{aP_a}{aP_a + bP_b + cP_c} = 0.0686 \tag{15}$$

短包到达率很低, 因而从时间统计来看, 短包抢占长包传输的概率就很小, 从而使整个系统的吞吐量下降很少. 通过仿真我们得到 P-SPF, P-MSM, P-MWM-QL 和 P-MWM-CA 都可取得 94% 的系统吞吐量. 同时, 由式 (14) 和式 (15) 可计算出系统吞吐量 T_{max} = 94%, 理论计算值和仿真结果是一致的.

图 6 给出了短包的平均包等待时间. 在已有包方式调度中, 短包的平均包等待时间随输入负载增加而迅速变大. 这主要是由于短包必须排队等在需要长服务时间的长包后面, 从而造成短包很大的平均包等待时间. 而在抢占模式下, 短包被分配了一个独立队列并可抢占长包的服务, 从而使得短包的平均包等待时间几乎降至零. 短包的服务时间仅为一个时隙, 因此短包总延迟也很小, 从而使得 TCP 中的 ACK 等控制包能更快地到达 TCP 的源端点.

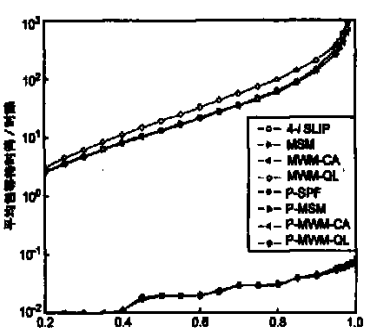

图 6 短包的平均包等待时间

图 7 给出了输入负载从 0.2 增加到 1.0 时所有包的平均包等待时间. 通过对比图中曲线可以看出: 已有包方式调度中, 4-iSLIP, MSM 和 MWM-CA 的性能几乎是相同的, 而 MWM-QL 的平均包等待时间最大. 在 MWM-QL 算法中: 当队列中长包较多时, 其权值较大, 该队列会优先得到服务; 反之, 当队列中短包较多时, 其权值较小, 该队列得到服务需要等待的时间就较长, 由此造成 MWM-QL 中短包平均包等待时间比较大. 同时短包占 50% 以上, 因此 MWM-QL 中所有包的平均包等待时间也会随之增大. 仿真中到达包的目的地址是在所有输出端口中均匀分布的, 所以任一 VOQ 队列总会在某个时隙有包到达. 当某 VOQ 队列长时间得不到服务时, 其队列长度就会不断增加并使其权值增大, 因此最终不会发生某队列永远得不到服务并导致整个系统不稳定的情况. 在抢占模式下, 所有四种算法的曲线几乎重叠在一起, 这表明简单易行的 P-SPF 可取得上述最大权重匹配算法的高性能. 从图 7 还可得出: 当输入负载小于 0.85 时, 抢占模式下的平均包等待时间小于已有包方式调度中的平均包等待时间; 当输入负载大于 0.85 时, 抢占模式的性能开始变差, 并在超过最大系统吞吐量 (94%) 后变得不稳定. Internet 中路由器的输入负载通常都小

于85%,因此抢占模式在实际网络中具有可实施性.

图8给出了P-SPF的抢占增益G,其中近乎水平的一条曲线来自式(11)的分析,另一条来自图7的仿真结果.在图8中可明显看出,当输入负载小于0.7时(低到中负载),分析的增益和仿真的增益是很匹配的,大约为2.0.当输入负载高于0.85时,两者之间的偏差增大.这是因为在高负载情况下,输入端口冲突成为了一个影响交换网络性能的不可忽略的因素.

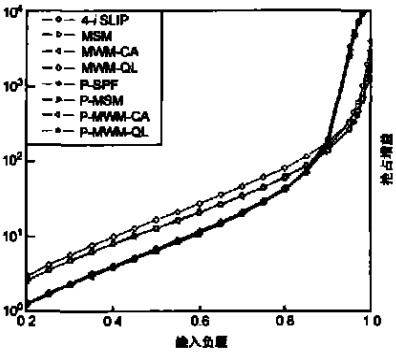

图7 所有包的平均包等待时间

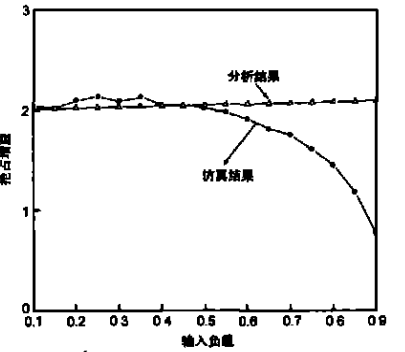

图8 P-SPF的抢占增益G

5 结论

本文通过研究输入排队交换结构中的包方式调度,指出一个有效的调度算法应考虑上层传输协议的特点.在对Internet业务特征进行分析后提出,TCP/IP网络中的短包应被确保高优先级和低延迟.以此为基础,我们研究了抢占模式下的包方式调度结构和P-SPF调度算法,并通过分析和仿真得到:P-SPF在所有输入负载下取得短包近似为零的平均包等待时间,在中低负载下降低所有包的平均包等待时间到原来的50%,并可取得94%的系统吞吐量.非常低的短包延迟带来的直接好处是:减小了TCP流的RTT时间,降低了TCP确认包超时的概率,从而实现更好的TCP传输性能.同时,由于基于RTP协议[22]或cRTP协议[23]的VoIP实时业务一般都采用短包传输,因此小的短包延迟也可使Internet提供更好的VoIP服务.

P-SPF调度算法同样可应用于IPv6核心路由器中.IPv6同IPv4的最大区别在于网络层的IP协议.IPv6的基本IP报头增加到了40字节,最小的TCP包也随之变为60字节.因此交换结构中的信元长度应适当增加,以使得大部分的TCP控制包只占一个信元,这样可减少IP包分割中由最后一个信元填充所造成的带宽浪费.IPv6在传输层仍采用现有TCP协议,其包长分布也将符合某种参数的TRIMODEL模型,因此本文中基于TCP的分析模型仍然适用.同时,IPv6中各个设备必须支持的MTU从576字节提高到了1280字节,网络中也允许出现大于64K字节的巨包,这就进一步使得长度相对较短的TCP控制包的阻塞概率增大,并最终导致TCP传输效率下降.因而在IPv6核心路由器中应用P-SPF可很好解决该问题,并可为用户提供更好的QoS.

参考文献:

[1] McKeown N. A fast switched backplane for a gigabit switched router [J]. Business Communications Review, 1997, 27(12): 1-30.

[2] McKeown N, Izzard M, Mekkittikul A, et al. Tiny tera: a packet switch core[J]. IEEE Micro, 1997, 17(1): 26-33.

[3] Partridge C, et al. A 50-Gb/s IP router[J]. IEEE/ACM Trans. Networking, 1998, 6(3): 237-248.

[4] Karol M J, Hluchyj M G, Morgan S. Input versus output queueing on a space-division packet switch[J]. IEEE Trans Commun, 1987, 35(12): 1347-1356.

[5] McKeown N. The *i*SLIP scheduling algorithm for input-queued switches [J]. IEEE/ACM Trans Networking, 1999, 7(2): 188-201.

[6] Mekkittikul A, McKeown N. A practical scheduling algorithm to achieve 100% throughput in input-queued switches[A]. IEEE INFOCOM 1998 [C]. San Francisco, USA: IEEE Computer and Communications Societies, 1998. 792-799.

[7] Chao H J. Saturn: a terabit packet switch using dual round robin[J]. IEEE Commun Mag, 2000, 38(12): 78-84.

[8] Marsan M A, Bianco A, Giaccone P, et al. Packet-mode scheduling in input-queued cell-based switches[J]. IEEE/ACM Trans. Networking, 2002, 10(5): 666-678.

[9] Ganjali Y, Keshavarzian A, Shah D. Input queued switches: cell switching vs. packet switching[A]. IEEE INFOCOM 2003[C]. San Francisco, USA: IEEE Computer and Communications Societies, 2003. 1651-1658.

[10] Zhang X, Bhuyan L N. Deficit round robin scheduling for input-queued switches[J]. IEEE J Sel Areas Commun, 2003, 21(4): 584-594.

[11] National Laboratory for Applied Network Research (NLANR). Auckland-II [EB/OL]. http://pma.nlanr.net/Special/.

[12] Wang R, Pau G, Yamada K, et al. TCP startup performance in large bandwidth delay networks[A]. IEEE INFOCOM 2004[C]. Hong Kong, China: IEEE Computer and Communications Societies, 2004. 796-805.

[13] Blake S, Black D, Carlson M, et al. An architecture for differentiated services[S]. IETF RFC 2475, 1998.

[14] Avrachenkov K E, Ayesta U, Brown P, et al. Differentiation between short and long TCP flows: predictability of the response time[A]. IEEE INFOCOM 2004[C]. Hong Kong, China: IEEE Computer and Communications Societies, 2004. 762-773.

[15] Rai I A, Keller G U, Vernon M, et al. Performance modeling of LAS based scheduling policies in packet switched networks[A]. ACM Sigmetrics-Performance 2004[C]. New York, USA: ACM SIGMETRICS, 2004. 106-117.

[16] Allen A O. Probability, statistics, and queueing theory with computer science applications[M]. New York: Academic Press, 1978.

[17] Baker F. Requirements for IP version 4 routers[S]. IETF RFC 1812, 1995.

[18] Bennett J C R, Partridge C, Shectman N. Packet reordering is not pathological network behavior[J]. IEEE/ACM Trans. Networking, 1999, 7(6): 789-798.

[19] Thompson K, Miller G J, Wilder R. Wide-area Internet traffic patterns and characteristics[J]. IEEE Network, 1997, 11(6): 10-23.

[20] Fraleigh C, et al. Packet-level traffic measurements from the sprint IP

backbone[J]. IEEE Network, 2003, 17(6): 6- 16.

[21] McKeown N, Anantharam V, Walrand J, Mekkittikul A. Achieving 100% throughput in an input-queued switch[J]. IEEE Trans Commun, 1999, 47(8): 1260- 1267.

[22] Casner S, Frederick R, Jacobson V, Schulzrinne H. RTP: A transport protocol for real-time applications[S]. IETF RFC 1889, 1996.

[23] Casner S, Jacobson V. Compressing IP/UDP/RTP headers for low-speed serial links[S]. IETF RFC 2508, 1999.

作者简介:

李文杰　男, 1978 年 11 月生于河南漯河, 博士研究生, 1996 年进入清华大学计算机科学与技术系读本科, 2000 年于该系计算机网络技术研究所直读博士, 研究方向为核心路由器体系结构, 高速交换技术, 调度算法, 以及宽带网 QoS 控制. E-mail: lwjie00@mails.tsinghua.edu.cn.

刘　斌　男, 1964 年 7 月生于山东临朐, 清华大学计算机科学与技术系教授, 博士生导师, 1993 年毕业于西北工业大学计算机应用专业, 获工学博士学位, 1993 年 5 月至 1995 年 11 月在北京邮电大学通信与电子系统专业从事博士后研究工作, 1995 年至今任教于清华大学, 科研方向为现代交换技术理论与方法学, 高速信息网络控制协议和性能评价, QoS 控制和多媒体业务量行为分析, 网络处理器及高速网络中的信息安全.